

# The Timestamp of Timed Automata

Amnon Rosenmann

Institute of Mathematical Structure Theory  
Graz University of Technology, Graz, Austria  
rosenmann@math.tugraz.at

## Abstract

Given a non-deterministic timed automaton with silent transitions (eNTA), we compute its timestamp: the set of all time values on which any observable transition occurs, and also a deterministic timed automaton with the same timestamp. The timestamp is eventually periodic and is constructed via a finite periodic augmented region automaton. A consequence of this construction is the periodicity of the language of timed automata with respect to suffixes. Applications include the decidability of the 1-bounded language inclusion problem for eNTA, and a partial method, not bounded by time or number of steps, for the general language non-inclusion problem for eNTA.

Keywords: timed automaton; timestamp of a timed automaton; periodic augmented region automaton; reachability problem; eventual periodicity

## 1 Introduction

Timed automata (TA) are finite automata extended with clocks that measure the time that elapsed since past events in order to control the triggering of future events. They were defined by Alur and Dill [AD94] as abstract models of real-time systems, and were implemented in tools like UPPAAL [LPY97], Kronos [BDM<sup>+</sup>98] and RED [Wan04].

A fundamental problem in this area is the reachability problem, where in its basic form it asks whether a given location of the TA is reachable from the initial location. The set of states of the system (i.e. locations and valuation to the clocks) is, in general, an infinite uncountable set. However, through the construction of the finitely-many equivalence classes of regions, the reachability problem becomes a simple decidable problem (although of complexity PSPACE-complete), as shown in [AD94].

Research on the reachability problem went beyond the above basic question. In [CY92] it is shown that the problem of the minimum and maximum reachability time is also PSPACE-complete. In another work, [CJ99], which is more

of a theoretical nature, the authors show that some problems on the relations between states may be defined in the decidable theory of the domain of real numbers equipped with the addition operation. In particular, the reachability problem between any two states is decidable. For other aspects of the reachability problem, also in the context of variants and extensions of timed automata (e.g. with game and probability characteristics) we refer to [CY92],[AKV98], [TY01], [WZP03], [AM99], [HP06], [CHKM11], [HOW12].

In this paper we generalize the reachability problem in another direction. We show that the problem of computing the set of all time values on which any observable transition occurs (and thus, a location is reached by an observable transition) is solvable. This set, called the *timestamp* of the automaton  $A$  and denoted  $\mathfrak{TS}(A)$ , is more precisely defined to be the set of all pairs  $(t, a)$  that appear in the observable timed traces of  $A$ . In other words, we transform each timed trace of  $A$  from a sequence of pairs  $(t, a)$  to a set of such pairs, and then take the union, over all timed traces, of these sets. Note that for this definition it does not matter whether we consider infinite runs or finite ones.

The timestamp is in the form of a union of action-labeled open intervals with integral end-points, and action-labeled points of integral values. When the timestamp is unbounded in time then it is eventually periodic: it has a time-prefix consisting of finitely-many labeled open unit intervals and labeled points of integral value, and then comes another finite such time-bounded pattern, which then repeats itself indefinitely. The length of the repeated pattern, i.e. the supremum on the set of differences between two time values in the same repeated pattern, is called a (timed) period of the automaton.

The set of language defined by the class DTA of deterministic timed automata is strictly included in the set of languages defined by the class NTA of non-deterministic timed automata [AD94], [Fin06], and the latter is strictly included in the set of languages defined by the class eNTA of non-deterministic timed automata with silent transitions [BPDG98]. The fundamental problem of the inclusion of the language accepted by one TA (e.g. the implementation) in the language accepted by another TA (e.g. the specification) is undecidable for NTA, but decidable for DTA. On the other hand, for special sub-classes or modifications it was shown that decidability exists (see [BBBB09, BPDG98, AFH99, BDFP04, OW04, ORW09, OW10, AM04] for a partial list). However, the abstraction (or over-approximation) represented in the form of a timestamp is a discrete object, in which questions like inclusion of timestamps or universality are decidable. In fact, we show that for any given eNTA one can construct a simple DTA having the same timestamp.

Applications of the timestamp include the decidability of the 1-bounded language inclusion problem for eNTA, and a partial method, which is not bounded by time or number of steps, for demonstrating a witness for the refutation of the general language inclusion problem for eNTA.

The computation of the timestamp is done through the construction of a periodic augmented region automaton. It is a region automaton augmented with a global non-resetting clock  $t$ , and which contains periodic regions and periodic transitions: they are defined modulo a time period  $L \in \mathbb{N}$ . If we

ignore time and leave only the actions then the periodic nature of the region automaton is evident from the cycles it contains. However, with continuous time there may only exist almost periodicity in single runs, not exact periodicity (see Example 6.1). Still, when we add the global time  $t$  to the region automaton in the more permissive manner, allowing periodic regions and periodic transitions, then it is possible to construct a finite automaton, in which time regions are part of the periods induced by the cycles in this automaton. Periodic transitions were introduced in [CG00], where it was shown that they increase the expressiveness of DTA, though they are less expressive than silent transitions.

We think that the periodic automaton is interesting in its own right since it bears an explicit demonstration of the property of time periodicity which characterizes eNTA. Moreover, if we restore the guards of the transitions of the periodic augmented region automaton then there is no loss of information with regard to the original automaton.

The construction of the periodic automaton is preceded by defining the infinite augmented region automaton, in which the values of the clock  $t$  are unbounded. Then, after exhibiting the existence of a pattern that repeats itself every  $L$  time units, we fold the infinite automaton into a finite one according to this periodic structure.

In terms of the language of an eNTA  $A$ , we show that it is periodic with respect to suffixes: for every run  $\varrho$  with suffix  $\varsigma$  that occurs after passing a fixed computable time there are infinitely-many runs of  $A$  with the same suffix  $\varsigma$ , but with the suffix shifted in time by multiples of  $L$ . Note that this result does not follow from the pumping lemma, which does not hold in general in timed automata [Bea98].

As for complexity, we do not seek here efficient algorithms for our constructions - this matter is postpone for future research. That is why we prefer to work over region automata, which are notorious for their inefficiency, but which make the analysis simpler.

In Section 2 basic definitions concerning timed automata are given. Then, in Section 3 we describe the trail and timestamp of a single path of a TA, more from a geometric than from an algebraic point of view, after treating the absolute global clock  $t$  as part of the system. The augmented and infinite augmented region automaton,  $\mathfrak{R}^t(A)$  and  $\mathfrak{R}_\infty^t(A)$ , are constructed in Section 4, and then, in Section 5, we explore the time-periodicity in them, so that  $\mathfrak{R}_\infty^t(A)$  can be folded into the finite periodic augmented region automaton  $\mathfrak{R}_{per}^t(A)$  (Section 6).

We started by exploring the timestamp of a single path of  $A$ , and in the last section (Section 7) we construct the entire eventually periodic timestamp. We also show that the 1-bounded universality and 1-bounded language inclusion problems are decidable for eNTA. As for the general language inclusion problem in eNTA, the timestamp, or better - the more informative  $\mathfrak{R}_{per}^t$ , may serve as a tool in demonstrating the non-inclusion relation between the languages of two eNTAs.

## 2 Timed Automata with Silent Transitions

A timed automaton (TA) is an abstract model aiming at capturing the temporal behavior of real-time systems. It is a finite automaton extended with a finite set of clocks defined over  $\mathbb{R}_{\geq 0}$ , the set of non-negative real numbers. The TA consisting of a finite set of *locations* with a finite set of *transitions* between the locations, while time, measured by the clocks, is continuous.

As long as the system is in some location  $q$ , all clocks advance at the same rate. Then at some point in time, the system may make a transition  $\tau$  to location  $q'$  (which may be  $q$  itself). Such a transition can occur if, first, the definition of the TA contains the transition  $\tau$  from  $q$  to  $q'$ . Secondly, the transition *guard*, which consists of a set of constraints in the form of strict or weak integer upper and/or lower bounds on the values of the clocks, should be satisfied at the time of the transition. The transition is immediate - no clock is advancing in time. However, some of the clocks may be reset to zero.

There are two sorts of transitions: *observable* transitions, which can be traced by an outside observer, and *silent* transitions, which are inner transitions and thus cannot be observed from the outside. There are finitely-many types of observable transitions, each type labeled by a unique *action*  $a \in \Sigma$ , whereas all the silent transitions have the same label  $\epsilon$ .

In NTA, non-deterministic TA, there exist states in which two transitions can be taken at the same time, with the same action, from the same location  $q$ , but to two different locations  $q'$  and  $q''$ . However, we would like to note that even in DTA, deterministic TA, at each time there may exist a choice of either to stay in the current location or to take a transition, and if a transition is taken there may be more than one possible (although with different actions).

A formal definition of eNTA is as follows.

**Definition 2.1** (eNTA). *A non-deterministic timed automaton with silent transitions, eNTA,  $A$  is a tuple  $(\mathcal{Q}, q_0, \Sigma_\epsilon, \mathcal{C}, \mathcal{T})$ , where:*

1.  $\mathcal{Q}$  is a finite set of locations and  $q_0$  is the initial location;
2.  $\Sigma_\epsilon = \Sigma \cup \{\epsilon\}$  is a finite set of transition labels, called actions, where  $\Sigma$  refers to the observable actions and  $\epsilon$  represents a silent transition;
3.  $\mathcal{C}$  is a finite set of clock variables;
4.  $\mathcal{T} \subseteq \mathcal{Q} \times \Sigma_\epsilon \times \mathcal{G} \times \mathcal{P}(\mathcal{C}) \times \mathcal{Q}$  is a finite set of transitions of the form  $(q, a, g, \mathcal{C}_{rst}, q')$ , where:
  - (a)  $q, q' \in \mathcal{Q}$  are the source and the target locations respectively;
  - (b)  $a \in \Sigma_\epsilon$  is the transition action;
  - (c)  $g \in \mathcal{G}$  is the transition guard, which is a conjunction of constraints of the form  $c \sim n$ , where  $c \in \mathcal{C}$ ,  $\sim \in \{<, \leq, =, \geq, >\}$  and  $n \in \mathbb{N}_0 := \mathbb{N} \cup \{0\}$ ;
  - (d)  $\mathcal{C}_{rst} \subseteq \mathcal{C}$  is the subset of clocks to be reset.

A clock *valuation*  $v(c)$  is a function  $v : \mathcal{C} \rightarrow \mathbb{R}_{\geq 0}$ . We denote by  $\mathcal{V}$  the set of all clock valuations and by  $\mathbf{0}$  the null valuation, which assigns the value 0 to every clock. Given a valuation  $v$  and  $d \in \mathbb{R}_{\geq 0}$ , we define  $v+d$  to be the valuation  $(v+d)(c) := v(c) + d$  for every  $c \in \mathcal{C}$ . The valuation  $v[\mathcal{C}_{rst}]$ ,  $\mathcal{C}_{rst} \subseteq \mathcal{C}$ , assigns the value 0 to every clock in  $\mathcal{C}_{rst}$  while leaving the other clocks unchanged.

The *semantics* of an eNTA  $A$  is given by the *timed transition system*  $[[A]] = (S, s_0, \mathbb{R}_{\geq 0}, \Sigma_\epsilon, T)$ , where:

1.  $S = \{(q, v) \in \mathcal{Q} \times \mathcal{V}\}$  is the set of states, with  $s_0 = (q_0, \mathbf{0})$  the initial state;
2.  $T \subseteq S \times (\Sigma_\epsilon \cup \mathbb{R}_{\geq 0}) \times S$  is the transition relation. Each  $T$  consists of
  - (a) *Timed transitions (delays)*:  $((q, v), d, (q, v+d)) \in T$ , where  $d \in \mathbb{R}_{\geq 0}$ ;
  - (b) *Discrete transitions (jumps)*:  $((q, v), a, (q', v')) \in T$ , where  $a \in \Sigma$  and there exists a transition  $(q, a, g, \mathcal{C}_{rst}, q')$  in  $\mathcal{T}$ , such that  $v \models g$  and  $v' = v[\mathcal{C}_{rst}]$ .

A (finite) *run*  $\rho$  of an eNTA  $A$  is a sequence of alternating timed and discrete transitions of the form

$$(q_0, \mathbf{0}) \xrightarrow{d_1} (q_0, d_1) \xrightarrow{\tau_1} (q_1, v_1) \xrightarrow{d_2} \dots \xrightarrow{d_k} (q_{k-1}, v_{k-1} + d_k) \xrightarrow{\tau_k} (q_k, v_k),$$

where  $\tau_i = (q_{i-1}, a_i, g_i, \mathcal{C}_{rst(i)}, q_i) \in \mathcal{T}$  and  $a_i \in \Sigma_\epsilon$ . The run  $\rho$  of  $A$  induces the *timed trace*

$$\lambda = (t_1, a_1), (t_2, a_2), \dots, (t_k, a_k),$$

with  $a_i \in \Sigma_\epsilon$  and  $t_i = \sum_{j=1}^i d_j$ . From the latter we can extract the *observable timed trace*, which is obtained by deleting from  $\lambda$  all the pairs containing silent transitions.

The definition of a *non-deterministic* timed automaton, denoted NTA, is similar to that of eNTA, except that it does not contain silent transitions. If, in addition, each timed trace induces a unique run we say that the TA is *deterministic*, denoted DTA.

**Remarks.** 1. it is common to include in the definition of a timed automaton a set of constraints called location invariants, which are upper bounds on the clocks values while staying in a location. However, for the purpose of this paper these constraints are not needed and we may consider them to be more of a ‘syntactic sugar’ because the invariants of location  $q$  may be incorporated in the guards of the transitions to  $q$  (for the clocks that are not reset at the transitions) and in those emerging from  $q$ .

2. The distinction between accepting and non-accepting locations, between observable and silent transitions, and between deterministic and non-deterministic automata is irrelevant to reachability problems. The language accepted by the automaton does depend on the observability of the transitions and the acceptance of the locations, however these restrictions do not matter for the analysis and results concerning the reachability problems that are presented in the paper.

### 3 The Trail and Timestamp of a Single Path

In this section we explore the trail and timestamp of a single path of a timed automaton. Given a timed automaton  $A \in \text{eNTA}$ , we add to it a non-resetting global clock  $t$ , which measures the absolute time. All clocks, including  $t$ , start with value 0 and then progress at the same rate.

**Definition 3.1** (Trajectory of a run). *Let  $\{t, x_1, \dots, x_r\}$  be an ordered set of clocks of an eNTA  $A$ . Let  $\varrho$  be a run of duration  $T$  of  $A$ . The trajectory of  $\varrho$  is the set of points  $(t, x_1, \dots, x_r)$  in the  $tx_1 \dots x_r$ -space visited during  $\varrho$ , where  $0 \leq t \leq T$ .*

**Definition 3.2** (Timestamp of a run). *The timestamp of a run  $\varrho$  is the finite set of pairs  $(t_i, a_i) \in \mathbb{R}_{\geq 0} \times \Sigma$  of the observable timed trace induced by  $\varrho$ .*

The timed automaton defines a graph, whose nodes represent the locations, and a directed edge connects a node representing location  $q$  with a node representing location  $q'$  if and only if there exists a transition  $\tau$  between  $q$  and  $q'$  in  $A$ . For simplicity, we use the same terminology of locations and transitions for nodes and edges, respectively, in the graph of  $A$ .

A finite path in the graph of  $A$  that starts at the initial location has the form  $\gamma = q_0 \tau_1 q_1 \tau_2 \dots \tau_n q_n$  of alternating locations and transitions. Such a path is an abstraction of a run since it does not contain the delays in the locations. Given a path  $\gamma$  in  $A$ , there may be many possible runs along  $\gamma$ , and we say that  $\gamma$  is *reachable* when there is at least one run along it.

**Definition 3.3** (Trail of a path). *The union of the trajectories of all runs along a path  $\gamma$  of  $A$  is called the trail of  $\gamma$ , denoted by  $\Theta_\gamma$ .*

**Definition 3.4** (Timestamp of a path). *The timestamp of a path  $\gamma$  of  $A$  is the union of the timestamps of all runs  $\varrho$  along  $\gamma$ .*

We call each instance of a transition along  $\gamma$  an *event*. That is, a transition is a static object which joins two locations of the timed automaton, whereas an event refers to a specific occurrence of a transition within the path  $\gamma$ . Hence, several events along a path may refer to the same transition of the timed automaton.

**Definition 3.5** (Timestamp of an event in a path). *The timestamp of an event in a path  $\gamma$  is the union of the timestamps of that event of all runs along  $\gamma$ . In other words, it is the part of the timestamp of the path that refers to that event.*

#### 3.1 Single Clock

We begin with a simple automaton  $A$  having a single clock  $x$  (in addition to  $t$ ). Let  $\gamma$  be a reachable path of  $A$  and let  $\varrho$  be a run along  $\gamma$ . The trajectory of  $\varrho$  represents the continuous evolution of the values of  $x$  during  $\varrho$  and it forms a

(discontinuous, in general) piecewise linear ‘function’.<sup>1</sup> which can be drawn in the  $tx$ -plane.

Let us now look at the set of all possible runs  $\varrho$  along  $\gamma$ . While each trajectory of a run is one-dimensional, the trail defined by all these runs may form a two-dimensional region of the  $tx$ -plane. A characteristic feature of the trail of a path when  $A$  has a single clock is that its width (the length of its intersection with a horizontal line) never decreases as time progresses.

The timestamp of  $\gamma$  is, in general, a union of labeled points and labeled intervals of  $\mathbb{R}_{\geq 0}$ . The timestamp of an event in  $\gamma$  is bounded between integral points, unless there is no upper bound, whereas for each run  $\varrho$ , its timestamp of the event consists of a single point.

**Example 3.1.** In Fig. 1(a) a TA is drawn, and in Fig. 1(b) we see the trail and timestamp of the path  $\gamma : (0) - (1) - (2) - (3) - (2)$ , where the timestamp of each action is given separately. The first event occurs when  $x = 1$  and the timestamp is  $\{1\} \times \{a\}$ . Then  $x$  resets and the trail (a straight line of slope 1) continues from the  $t$ -axis. Event 2 occurs when  $1 \leq x \leq 3$  with timestamp  $[2, 4] \times \{b\}$  and a reset of  $x$ . After that event the trail is two-dimensional (a parallelogram). Event 3 occurs when  $1 < x < 2$  without clock reset, and the orthogonal projection to the  $t$ -axis gives the timestamp  $(3, 6) \times \{a\}$  (here  $(3, 6)$  is the open interval  $3 < t < 6$ ). The fourth event happens when  $x = 3$  and its timestamp is  $[5, 7] \times \{a\}$ . The timestamp of  $\gamma$  is the union of the above sets, that is,  $S_1 \times \{a\} \cup S_2 \times \{b\}$ , with  $S_1 = \{1\} \cup (3, 7]$  and  $S_2 = [2, 4]$ .

The trail  $\Theta_\gamma$  can be decomposed into diagonal lines and parallelograms of width 1, and each diagonal line or parallelogram can be further decomposed into basic shapes called *regions* [AD94] that come in periodic manner and consist of open triangles, open lines and points (see Fig. 1(c)).

### 3.1.1 The Width of a Trail

In the next proposition we give a recursive formula for the non-decreasing sequence of widths of the trail of a path. Given a path  $\gamma$  in  $A$ , we denote by  $d_i^\gamma$  the feasible *duration* of the  $i$ -th event along  $\gamma$  (the length of time between the possible earliest and latest readings on clock  $x$  when taking the  $i$ -th transition). The values  $d_i$  are computed in the following way. Let  $l_i^\gamma, u_i^\gamma$  be the lower and upper bound of the guard on  $x$  at the  $i$ -th transition along  $\gamma$  ( $l_i^\gamma = u_i^\gamma$  in case of an exact time event). When the lower bound is missing it is set to 0, and when the upper bound is missing it is set to  $\infty$ . Then, starting from the first event and proceeding forward, for each  $i$ , if  $x$  is not reset at event  $i$  and if  $l_{i+1}^\gamma < l_i^\gamma$  then we set  $l_{i+1}^\gamma := l_i^\gamma$ . Similarly, starting from the last event of  $\gamma$  and proceeding backward, for each  $i$ , if  $x$  is not reset at event  $i$  and if  $u_i^\gamma > u_{i+1}^\gamma$  then we set  $u_i^\gamma := u_{i+1}^\gamma$ . Then we define  $d_i^\gamma := \max\{u_i^\gamma - l_i^\gamma, 0\}$ . Let also  $s_i^\gamma$  be the size of the timestamp of the  $i$ -th event, and let  $w_i^\gamma$  be the width of the trail after the  $i$ -th event, with  $w_0^\gamma$  the initial width of the trail.

<sup>1</sup>In fact, at the points of discontinuity (when  $x$  is reset) we have two values of  $x$  for the same value of  $t$ .

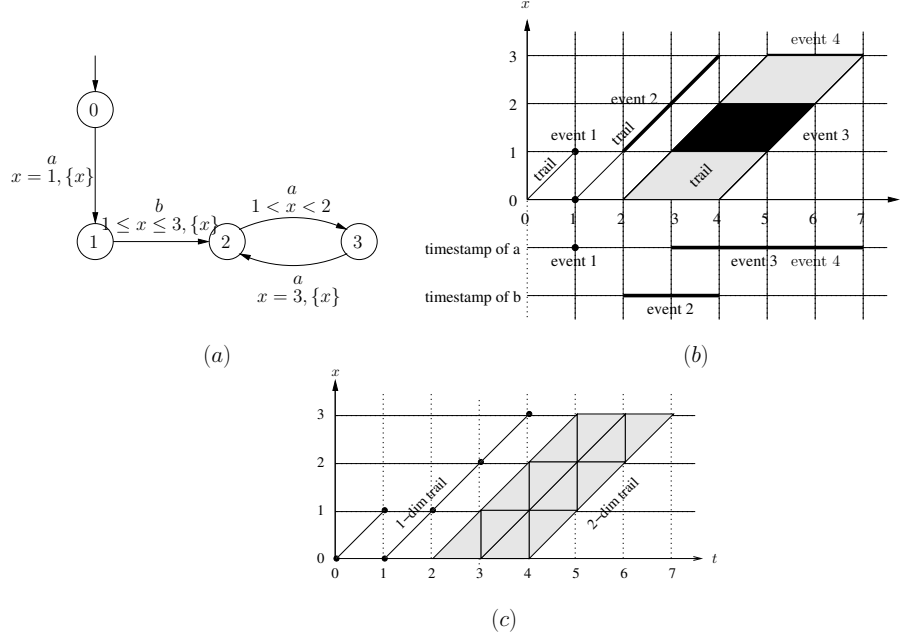


Figure 1: Trail, timestamp and regions of a path (single clock)

**Proposition 3.1.** *Let  $\gamma$  be a reachable path in  $A$ . Then*

$$s_i^\gamma = w_{i-1}^\gamma + d_i^\gamma \quad (1)$$

and

$$w_i^\gamma = \begin{cases} 0 & \text{if } i = 0 \\ s_i^\gamma & \text{if } i > 0 \text{ and } x \text{ is reset on event } i \\ w_{i-1}^\gamma & \text{otherwise} \end{cases}$$

*Proof.* When  $x$  is not reset on an event then the trail continues as before the event and certainly its width does not change:  $w_i^\gamma = w_{i-1}^\gamma$ . If  $x$  is reset then the size of the timestamp of the event is the size of the interval in the  $t$ -axis which consists of all points  $(t, 0)$  of the trail after the reset. But after the event the trail contains all the parallel trajectories that pass through these points  $(t, 0)$  and so the new width of the trail is  $w_i^\gamma = s_i^\gamma$ .

Let us look now at the size of the timestamp of the  $i$ -th event. Suppose it refers to the actual constraint  $m < x < m + d_i^\gamma$  (the strict inequalities may be replaced by weaker ones). Let  $t_0 = \inf_t \{(t, m)\}$ , where  $(t, m)$  is in the part of the trail after the  $(i-1)$ -th event. Then the event refers to the parallelogram (or to a line, in the degenerate case) with vertices at  $(t_0, m)$ ,  $(t_0 + w_{i-1}^\gamma, m)$ ,  $(t_0 + d_i^\gamma, m + d_i^\gamma)$ ,  $(t_0 + w_{i-1}^\gamma + d_i^\gamma, m + d_i^\gamma)$ , and the earliest time at which the event can occur is  $t_0$ , while the latest time is  $t_0 + w_{i-1}^\gamma + d_i^\gamma$ . Thus,  $s_i^\gamma = w_{i-1}^\gamma + d_i^\gamma$ .  $\square$



### 3.2 Multiple Clocks

Assume that  $A$  has  $r > 1$  clocks  $x_1, \dots, x_r$ . Let  $\gamma$  be a reachable path in the graph of  $A$ . The trajectory of a run  $\varrho$ , consisting of all points  $(t, x_1, \dots, x_r)$  during  $\varrho$ , forms a (discontinuous, in general) piecewise linear curve in the  $tx_1 \cdots x_r$ -space.

The trail of  $\gamma$ ,  $\Theta_\gamma$ , is the union of the trajectories of all runs along  $\gamma$ . The parts of the trail between clocks reset are called *zones* [DT98]. Each zone can be partitioned into *simplicial trails*, which are (possibly unbounded) parallelotopes of width (i.e. the length of the intersection with the line  $x_1 = c_1, \dots, x_r = c_r$ ) is 1 or 0, and then each simplicial trail can be triangulated into regions.

In the rest of this subsection we denote the global clock  $t$  by  $x_0$  for simplicity of presentation. Each region  $\mathbf{n} + \Delta$  is in the form of an open (unless it is a point) simplex  $\Delta$  of dimension 0 (a point), 1 (a line), 2 (a triangle), 3 (a tetrahedron) or higher, up to dimension  $r + 1$ , that resides inside an  $(r + 1)$ -dimensional unit hyper-cube with the vertices of the simplex in the lattice  $\mathbb{N}_0^{r+1}$ . The simplex  $\Delta$  is characterized by the fractional values  $\{x_i\}$  of the clock variables, and each point in the simplex satisfies the same fixed ordering of the form  $0 \preceq_1 \{x_{i_1}\} \preceq_2 \{x_{i_2}\} \preceq_3 \cdots \preceq_r \{x_{i_r}\}$ ,  $\preceq_i \in \{=, <\}$ . The integral point  $\mathbf{n} \in \mathbb{N}_0^{r+1}$  consists of the integral parts of the values of the clocks  $x_0, x_1, \dots, x_r$ , and it indicates the lowest point in the  $tx_1 \cdots x_r$ -space of the boundary of the region.

Each simplicial trail  $S$  is of the form  $\mathbf{n} + \Delta + \{\alpha \mathbf{1} \mid \alpha \geq 0\}$ , where  $\mathbf{n} + \Delta$  is a region that is obtained after an event, and then this simplex is ‘moved’ along the directional vector  $\mathbf{1} = (1, 1, \dots, 1)$ . When  $S$  is  $k$ -dimensional then the simplices along it (in the  $\mathbf{1}$ -direction) are switched alternately between an open  $k$ -simplex (a simplex of dimension  $k$ ,  $1 \leq k \leq r + 1$ ) and a  $(k - 1)$ -simplex, where each  $(k - 1)$ -simplex is a face of its neighboring  $k$ -simplices, and each switch is characterized by a cyclic shift in the fractional parts of the clocks. As with a single clock, the sequence of simplices along the simplicial trail is periodic.

However, unlike the situation with a single clock, the dimension of the trail can increase, decrease or stay the same after an event with reset of clocks: clocks with the same fractional part can be separated, resulting in an increase of the dimension, while clocks whose fractional parts become identical (namely, 0) contribute to a decrease of the dimension.

**Proposition 3.2.** *The timestamp of each event involving multiple clocks is either a labeled integral point or a labeled (open, closed or half-open) interval between points  $m$  and  $n$ ,  $m < n$ ,  $m \in \mathbb{N}_0$  and  $n \in \mathbb{N} \cup \infty$ .*

*Proof.* At the initial state all clocks are set to 0 and then progress at the same rate, i.e. the trail forms a straight line in direction  $\mathbf{1} = (1, 1, \dots, 1)$  in the  $tx_1 \cdots x_r$ -space. Suppose, by induction, that after the  $i$ -th event the trail has the form of a parallelotope leaning in direction  $\mathbf{1}$ , which is decomposed into simplices as described above that lie on the  $\mathbb{N}_0^{r+1}$  lattice.

The guard of the transition of the  $(i + 1)$ -th event is the conjunction of constraints of the form  $x_i = n_i$  (a hyperplane) or of the form  $x_i < n_i$  or  $x_i > n_i$  (a half-space). The intersection of these domains is a convex set parallel (or

orthogonal) to the axes and with faces defined by integer values. The intersection with the trail, the event domain, is again a convex set which is union of simplices as above. Consequently, the unlabeled timestamp of the event, which is the orthogonal projection to the  $t$ -axis of the event domain, is a convex set with integer end-points (unless unbounded above).

If no clock is reset at  $(i + 1)$ -th event then the trail has the same form as before the transition. Otherwise, the projection of the event domain is again a convex set, which is the union of simplices lying on the  $\mathbb{N}_0^{r+1}$  lattice. From that time on, the projected event domain is ‘moved’ in direction  $\mathbf{1}$  and thus it is again a parallelotope leaning in direction  $\mathbf{1}$ , and, by induction, the timestamp of each of the following events is of the same form as stated in the claim.  $\square$

## 4 The Augmented and Infinite Augmented Region Automaton

The region automaton  $\mathfrak{R}(A)$  [AD94] is a finite automaton in which time is abstracted, and such that  $A$  and  $\mathfrak{R}(A)$  define the same untimed language. Each node in the graph representation of  $\mathfrak{R}(A)$  records a location in  $A$  and a region, which is either a simplex (as described in the previous section) or an unbounded region in which at least one of the clocks passed the maximal integer value  $M$  that appears in the guards of the automaton. The edges of  $\mathfrak{R}(A)$  are labeled by the transition actions. We remark that in order to restore the information of the original automaton  $A$  one can add the transition guards to the edges.

We extend now  $\mathfrak{R}(A)$  by adding to it the global clock  $t$ . However, unlike the other clocks, we consider all integral values of  $t$  and not only the values  $0, 1, \dots, M$ . That is, the time regions associated with the continuous time  $t$  are the alternating point and open unit interval:  $\{0\}$ ,  $(0, 1)$ ,  $\{1\}$ ,  $(1, 2)$ ,  $\{2\}$ ,  $(2, 3)$ ,  $\dots$ , and the region automaton is unfolded with respect to these time regions. The result is the *infinite augmented region automaton*, denoted by  $\mathfrak{R}_\infty^t(A)$ . Each node of  $\mathfrak{R}_\infty^t(A)$  is a triple

$$(q, \mathbf{n}, \Delta), \tag{2}$$

where  $q$  is a location of  $A$ ,  $\Delta$  is the simplex of the fractional parts of the clocks  $t, x_1, \dots, x_r$ , and

$$\mathbf{n} = (n_0, n_1, \dots, n_r) \in \mathbb{N}_0 \times \{0, 1, \dots, M, \top\}^r \tag{3}$$

contains the integral parts of the clocks. When the value of a regular clock  $x_i$  passes  $M$  then its value is represented by the symbol  $\top$ . As we see, the structure of  $\mathfrak{R}_\infty^t(A)$  is the similar to that of the region automaton, except for the additional clock  $t$ , which makes it, in general, infinite. Note that an edge of  $\mathfrak{R}(A)$  that enters a region  $D$  in a transition on which all regular clocks have value greater than  $M$  (some clocks may be reset during the transition) is represented in  $\mathfrak{R}_\infty^t(A)$  by infinitely-many edges since  $D$  is split into infinitely-many regions

$D_i$  of the different time regions. Similarly for edges going out of a region  $D$  with all regular clocks of value greater than  $M$ .

A second construction is the the finite *augmented region automaton*, denoted by  $\mathfrak{R}^t(A)$ . Here we consider only the fractional part of  $t$  and ignore its integral part, that is, we represent  $t$  modulo 1. Thus,  $\mathfrak{R}^t(A)$  can be obtained by folding  $\mathfrak{R}_\infty^t(A)$  and identifying nodes which only differ by the integral part of  $t$ . The compensation for this loss of information is gained by assigning each edge of  $\mathfrak{R}^t(A)$  a non-negative integral *weight*, which is the difference  $\lfloor t_1 \rfloor - \lfloor t_0 \rfloor$  between the integral parts of the values of  $t$  in the target and the source nodes. When there is no bound to this difference (by a transition to or from a region  $D$  as above) then we assign such an edge the value  $m^*$ , where  $m$  is the *minimal* possible integer difference in the values of  $t$  between the target and the source nodes.

**Example 4.1.** In Fig. 2(a) we see a TA  $A$  containing a transition to an unbounded region. The corresponding infinite augmented region automaton  $\mathfrak{R}_\infty^t(A)$  is shown in Fig. 2(b). Each node of  $\mathfrak{R}_\infty^t(A)$  is represented by a rounded rectangle containing the original location of  $A$  (encircled, on the left), the integral values of  $t$  and of  $x$  (in the top of the rectangle) and the simplex (in the bottom). Notice that when the value of  $x$  is greater than  $M = 0$  it is marked by  $\top$  and its fractional part is ignored. To the left of  $\mathfrak{R}_\infty^t(A)$  we see the discretization of time  $t$  into time regions, and each node of  $\mathfrak{R}_\infty^t(A)$  is drawn in the level of its time region. In Fig. 2(c) the augmented region automaton  $\mathfrak{R}^t(A)$  is shown. Here the integral part of the value of  $t$  is ignored. The edge labeled by  $0^*$  represents the infinitely-many differences in the integral parts of the values of  $t$ :  $0, 1, 2, \dots$ . Similarly, the edge labeled with  $1^*$  refers to the differences  $1, 2, 3, \dots$ .

## 5 Eventual Periodicity

In this section we address the main topic of this paper: exploring the time-periodic property of a TA. In addition to demonstrating its existence, we show how one can actually compute the parameters of a period.

### 5.1 Cycles without Zenoness in $\mathfrak{R}^t(A)$

The graph  $\mathfrak{R}^t(A)$  is a finite connected directed graph with an initial node. In what follows, a ‘path’ in  $\mathfrak{R}^t(A)$  always starts at the initial node  $g_0$ , unless otherwise stated. Note that every edge of  $\mathfrak{R}^t(A)$  corresponds to a transition in  $A$  that can be taken.

**Definition 5.1** (Duration of a path). *Given a path  $\gamma$  in  $\mathfrak{R}^t(A)$ , its minimal integral duration, or simply duration,  $d(\gamma) \in \mathbb{N}_0$  is the sum of the weights on its edges, where a weight  $m^*$  is counted as  $m$ .*

**Definition 5.2** (Zenoness of a Cycle). *A cycle of  $\mathfrak{R}^t(A)$  has the Zenoness property if all its edges have weight 0.*

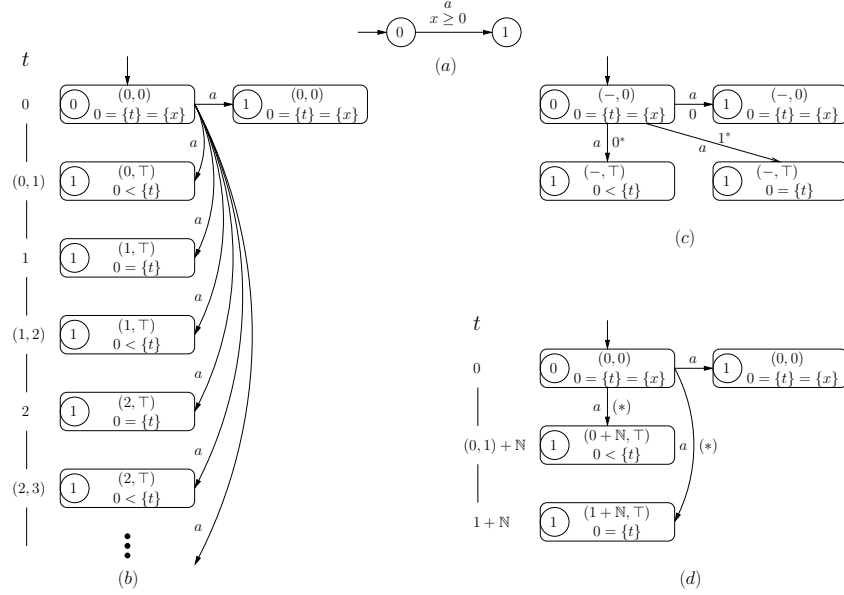


Figure 2: (a)  $A \in \text{TA}$ ; (b) The infinite augmented region automaton  $\mathfrak{R}_\infty^t(A)$ ; (c) The augmented region automaton  $\mathfrak{R}^t(A)$ ; (d) A periodic augmented region automaton  $\mathfrak{R}_{per}^t(A)$

**Lemma 5.1.** *There exists a computable positive integer  $t_0$ , such that every path  $\gamma$  of  $\mathfrak{R}^t(A)$  that is of (minimal) duration  $t_0$  or more contains a node belonging to some cycle without Zenoness.*

*Proof.* If there is a bound on the duration of all paths of  $\mathfrak{R}^t(A)$  then the claim holds trivially. Otherwise, let  $N$  be the number of nodes in  $\mathfrak{R}^t(A)$  and let  $\gamma$  be a path of duration greater than  $MN$ , where  $M$  is the maximal weight of an edge. Then  $\gamma$  must contain cycles. But if a cycle has the Zenoness property then its duration is 0. Thus, if all cycles in  $\gamma$  were with Zenoness then an increase in the accumulative duration of the path could have happened only when passing an edge whose end node is visited for the first time and then the duration of  $\gamma$  would have been at most  $MN$  - a contradiction.  $\square$

## 5.2 A Period of $\mathfrak{R}^t(A)$

Recall that a path in a graph is a simple cycle if its nodes are pairwise distinct, except for the first and last one. We are interested here in cycles without Zenoness, so let  $\bar{\mathfrak{R}}^t(A)$  be the graph obtained from  $\mathfrak{R}^t(A)$  by contracting each cycle with Zenoness to a single node.

**Definition 5.3** (Level of a simple cycle). *The level of a simple cycle of  $\bar{\mathfrak{R}}^t(A)$  is the minimal number of simple cycles visited on a path (from the initial node)*

that reaches it. In particular, a simple cycle  $\pi$  is of level 1 if there exists a path whose terminating node belongs to  $\pi$  and all other nodes do not belong to any simple cycle.

**Definition 5.4** (Cycle of type wZ1). A cycle of  $\mathfrak{R}^t(A)$  is of type wZ1 if its image in the graph  $\mathfrak{R}^t(A)$  is a simple cycle of level 1.

Let  $P$  be a minimal node cover of the collection of cycles of type wZ1 of  $\mathfrak{R}^t(A)$ . That is:

1. The set of nodes in the cycles in  $P$  equals the set of nodes in all cycles of type wZ1;
2. Each cycle in  $P$  contains a node (at least one) that does not belong to any of the other cycles in  $P$ .

For each cycle  $\pi$  in  $P$  let  $d(\pi)$  be its duration.

**Definition 5.5** (Period of  $\mathfrak{R}^t(A)$ ). A time period, or just period,  $L = L(\mathfrak{R}^t(A))$  of the augmented region automaton  $\mathfrak{R}^t(A)$  is the least common multiple, which is also greater than  $M$ , of the set  $\{d(\pi) \mid \pi \in P\}$ . If  $P$  is empty then  $L$  is set to be 1.

That is, if  $l$ , the least common multiple of the integers  $d(\pi)$  is greater than  $M$  then  $L$  is defined to be  $l$ . Otherwise,  $L$  is defined to be the smallest multiple of  $l$  which is greater than  $M$ . We would like to remark that the logic behind the above definition is to reduce the number of cycles involved in the definition of  $L$ , possibly resulting in a smaller period. However, a more thorough exploration of the duration of cycles, e.g. taking into account their common factors, would probably lead to a smaller period.

### 5.3 Eventual Periodicity of $\mathfrak{R}_\infty^t(A)$

**Lemma 5.2.** If  $\mathfrak{R}_\infty^t(A)$  is not bounded in time then there exists a computable positive integer  $t_0$ , such that for each edge  $e$  of  $\mathfrak{R}_\infty^t(A)$  of the form

$$[g = (q, (n_0, n_1, \dots, n_r), \Delta) \xrightarrow{a} (q', (n'_0, n'_1, \dots, n'_r), \Delta') = g'], \quad (4)$$

with  $n_0 \geq t_0$ , there exists an edge  $e'$  of  $\mathfrak{R}_\infty^t(A)$  of the form

$$[h = (q, (n_0 + L, n_1, \dots, n_r), \Delta) \xrightarrow{a} (q', (n'_0 + L, n'_1, \dots, n'_r), \Delta') = h']. \quad (5)$$

*Proof.* We take  $t_0$  to be the integer of Lemma 5.1. Let  $\gamma$  be a path of  $\mathfrak{R}_\infty^t(A)$  which terminates in the node  $g$  as above. Let  $\gamma' = p(\gamma)$  be the image of  $\gamma$  under the projection to  $\mathfrak{R}^t(A)$ . If  $\gamma$  contains an edge  $e_1$  such that  $e'_1 = p(e_1)$  has unbounded weight  $m^*$  then we can replace  $e_1$  by another edge of  $p^{-1}(e'_1)$  with a delay that is greater than the delay of  $e_1$  by  $L$ . Otherwise, since  $n_0 \geq t_0$  then, by Lemma 5.1,  $\gamma'$  contains a node which belongs to a cycle without Zenoness, and consequently a node  $g''$  of a cycle  $\pi \in P$  of type wZ1. A variant of the pumping lemma finishes the proof by enlarging  $\gamma$  with  $L/d(\pi)$  cycles of  $\pi$  at  $g''$ .  $\square$

As a consequence we get the following.

**Theorem 5.3.** *The infinite augmented region automaton  $\mathfrak{R}_\infty^t(A)$  is eventually periodic: there exists an integral time  $t_{\text{per}} > 0$  and an integer  $L > 0$ , such that for each  $t \geq t_{\text{per}}$ , the subgraph of  $\mathfrak{R}_\infty^t(A)$  that starts at time-level  $t$  is isomorphic, up to an  $L$ -shift in time, to the subgraph that starts at time-level  $t + L$ .*

*Proof.* For each  $k = 0, 1, 2, 3, \dots$  let  $\bar{G}_k$  be the set of nodes of  $\mathfrak{R}_\infty^t(A)$  in which the integral part of clock  $t$  satisfies  $t_0 + kL \leq \lfloor t \rfloor < t_0 + (k+1)L$ , where  $t_0$  is as in Lemma 5.1. By Lemma 5.2,  $\bar{G}_0 \subseteq \bar{G}_1 \subseteq \bar{G}_2 \subseteq \dots$ . But there is a bound on the number of possible different nodes of  $\bar{G}_k$  since  $t$  is evaluated modulo  $L$ , hence the sequence  $\bar{G}_k$  eventually stabilizes. The result then follows since the set of out-going edges from each node does not rely on the value of  $t$  but on the values of the other clocks.  $\square$

The time  $t_{\text{per}}$  is computable by the following lemma, which says that we can stop constructing the graph when the sequence  $\bar{G}_k$  does not increase two consecutive times.

**Lemma 5.4.** *if  $\bar{G}_k = \bar{G}_{k+1} = \bar{G}_{k+2}$  then  $\bar{G}_k = \bar{G}_{k+j}$  for every  $j \geq 1$ .*

*Proof.* Suppose that  $\bar{G}_k = \bar{G}_{k+1} = \bar{G}_{k+2}$ . By induction, it suffices to show that  $\bar{G}_{k+2} = \bar{G}_{k+3}$ . Let  $g$  be a node in  $\bar{G}_{k+3}$ . We need to find an equivalent node  $g' \in \bar{G}_{k+2}$ . If  $g$  is reached by an edge from a node in  $\bar{G}_{k+1}$  or in  $\bar{G}_{k+2}$  then an equivalent node  $g' \in \bar{G}_{k+2}$  is reached from a node in  $\bar{G}_k$  or in  $\bar{G}_{k+1}$  respectively. Otherwise,  $g$  is reached by an edge  $e$  from a node  $h$  in  $\bar{G}_k$  or earlier, and the time difference  $d$  between  $h$  and  $g$  is greater than  $2L$ . This implies that the projection  $p(e) \in \mathfrak{R}^t(A)$  of  $e$  is of unbounded time delay  $m^*$ . Since  $L > M$  then  $d - L > M$ . Hence, there is another edge  $e'$  in  $\mathfrak{R}_\infty^t(A)$ , that is also a pre-image of  $p(e)$ , and which joins  $h$  to a node  $g' \in \bar{G}_{k+2}$ , where  $g'$  is equivalent to  $g$ .  $\square$

**Example 5.1.** *This example refers to the timed automaton of Fig. 4 (a). In order to make the analysis of its time-periodic structure simpler, we changed the guard on the transition from location 1 to location 2 to be simpler (Fig. 3 (a)), so that in the resulting infinite augmented region automaton  $\mathfrak{R}_\infty^t(A)$  (Fig. 3 (b)) we can clearly see two different cycles of period 6 (encircled in dotted lines) (the edges with label  $c$  are only partly shown). We then added the original guard between locations 1 and 2 (Fig. 4 (a)). In the additional part in the graph of  $\mathfrak{R}_\infty^t(A)$  (Fig. 4 (b)) we see two more cycles, one of period 11 and one of period 5. We can still use a period of length 6 for this more complex automaton, but the existence of cycles of other lengths results in a longer time until reaching the repeated periodic part of the entire automaton.*

The time periodicity of  $\mathfrak{R}_\infty^t(A)$  is translated into the periodicity of the language of eNTA  $A$  with respect to suffixes.

**Theorem 5.5.** *The language of  $A \in \text{eNTA}$  is suffix-periodic: there exists time  $t_{\text{per}} > 0$  and integer  $L > 0$ , such that if  $t_r > t_{\text{per}}$  and*

$$\lambda = (t_1, a_1), \dots, (t_{r-1}, a_{r-1}), (t_r, a_r), (t_{r+1}, a_{r+1}), \dots, (t_{r+m}, a_{r+m})$$

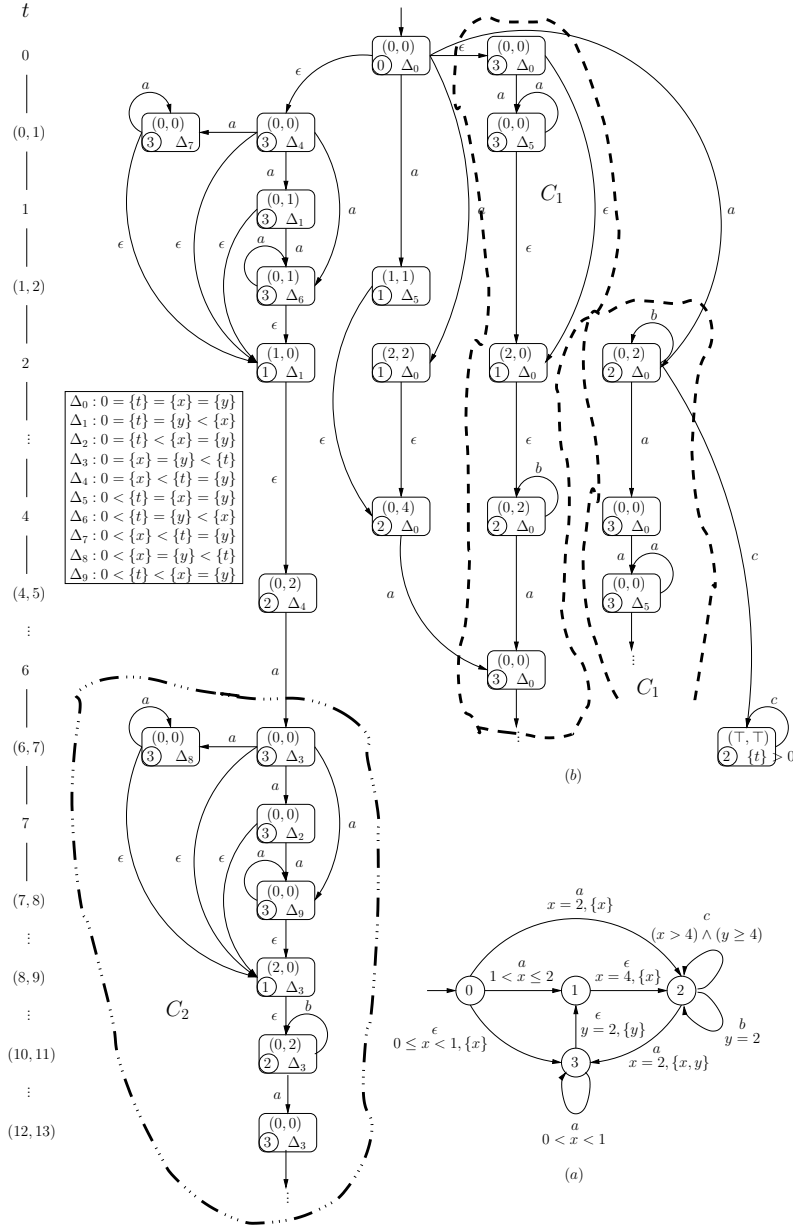


Figure 3: (a) The simplified  $A \in \text{eNTA}$ ; (b)  $\mathfrak{R}_\infty^t(A)$  with period 6

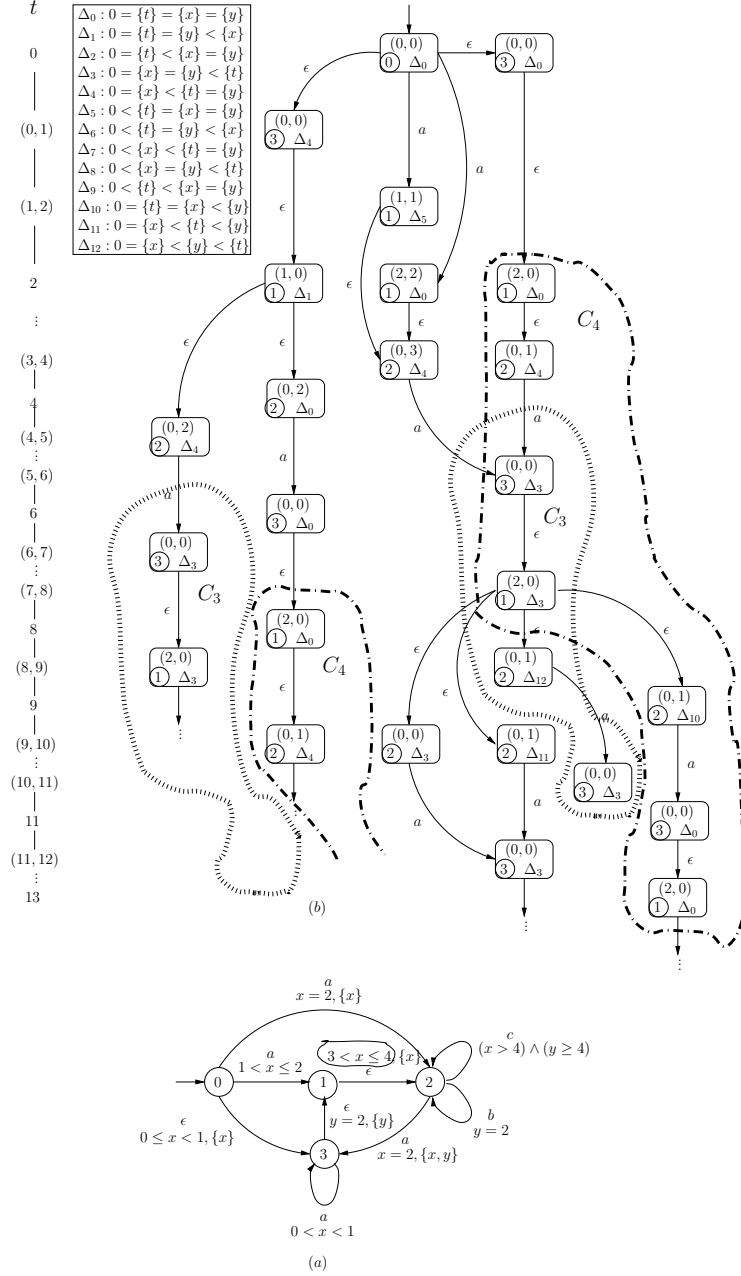


Figure 4: (a) The original  $A \in \text{eNTA}$ ; (b) The additional part of  $\mathfrak{R}_\infty^t(A)$  with cycles of lengths 11 and 5



is an observable timed trace of  $\mathfrak{L}(A)$  then, for each  $K \in L\mathbb{Z}$ , if  $t_r + K > t_{per}$  then there exists an observable timed trace  $\lambda' \in \mathfrak{L}(A)$  such that

$$\lambda' = (t'_1, a'_1), \dots, (t'_s, a'_s), (t_r + K, a_r), (t_{r+1} + K, a_{r+1}), \dots, (t_{r+m} + K, a_{r+m}).$$

*Proof.* If  $A$  is bounded in time then the claim holds vacuously. Otherwise, suppose that  $\lambda$  is obtained by some run  $\varrho$  through  $A$ , and that at time  $t_r$  it reaches location  $q$ . This run corresponds to a path in  $\mathfrak{R}_\infty^t(A)$ , whose  $r$ -th transition reaches a node  $g$  with the time region of  $t_r$ . By Theorem 5.3, there exists a path in  $\mathfrak{R}_\infty^t(A)$  which reaches an equivalent node  $g'$  with the time region of  $t_r + K$  (with the same location  $q$  and the same simplex with regard to the fractional parts of the clocks). Then, by Proposition 3.2 and its proof, if a region is reached at some time then it can be reached at any other time within the same region, and the result then follows.  $\square$

## 6 A Periodic Augmented Region Automaton

After revealing the periodic structure in  $\mathfrak{R}_\infty^t(A)$ , it is natural to fold it into a finite graph, called a *periodic augmented region automaton* and denoted by  $\mathfrak{R}_{per}^t(A)$ . The construction of  $\mathfrak{R}_{per}^t(A)$  is done in the following way. We start with the subgraph of  $\mathfrak{R}_\infty^t(A)$  of time  $t < t_{per} + L$ . Then, each node  $g$  of time  $t_{per} \leq t < t_{per} + L$  (the periodic subgraph) is turned into a *periodic* node with the integral value of  $t$  written as  $n + L\mathbb{N}_0$ ,  $t_{per} \leq n < t_{per} + L$ , referring to all nodes of integral time  $n + kL$ ,  $k = 0, 1, 2, \dots$  of  $\mathfrak{R}_\infty^t(A)$ . The only additional modification is to mark by a star (\*) the edges where the values of all regular clocks are greater than  $M$ , indicating that this edge represents infinitely-many edges stemming from a single node of  $\mathfrak{R}_\infty^t(A)$  and terminating in infinitely many nodes of  $\mathfrak{R}_\infty^t(A)$ , all of the same time modulo  $L$ .

In general, an (unmarked or marked) edge of the periodic part of  $\mathfrak{R}_{per}^t(A)$  corresponds to infinitely-many edges of  $\mathfrak{R}_\infty^t(A)$  with infinitely-many source nodes but with each edge having a single target node. If the edge  $e$  of  $\mathfrak{R}_{per}^t(A)$  joins a node of integral time  $n_1 + L\mathbb{N}_0$  with a node of integral time  $n_2 + L\mathbb{N}_0$ ,  $t_{per} \leq n_1, n_2 < t_{per} + L$ , then the following holds. If  $n_1 \leq n_2$  then  $e$  represents the infinitely-many edges of  $\mathfrak{R}_\infty^t(A)$  connecting the corresponding nodes of integral times:  $n_1 \rightarrow n_2$ ,  $n_1 + L \rightarrow n_2 + L$ ,  $n_1 + 2L \rightarrow n_2 + 2L$ , and so on. If  $n_1 > n_2$  then the corresponding edges of  $\mathfrak{R}_\infty^t(A)$  are:  $n_1 \rightarrow n_2 + L$ ,  $n_1 + L \rightarrow n_2 + 2L$ ,  $n_1 + 2L \rightarrow n_2 + 3L$ , and so on.

In Fig. 2 we see the four different representations of a very simple automaton  $A$ , with Fig. 2(d) being the periodic augmented region automaton.

**Example 6.1.** The timed automaton shown in Fig. 5(a) is taken from [AD94], where it is used for the purpose of demonstrating non-periodicity as the time difference between an  $a$ -transition and the following  $b$ -transition is strictly decreasing. However, the periodicity among the collection of timed traces is seen in the periodic augmented automaton, where the period here is of size 1, and the nodes in times  $(2, 3) + \mathbb{N}_0$  and  $3 + \mathbb{N}_0$  are periodic. Notice also that there

are edges marked with  $(*)$  which represent infinitely-many edges with the same source.

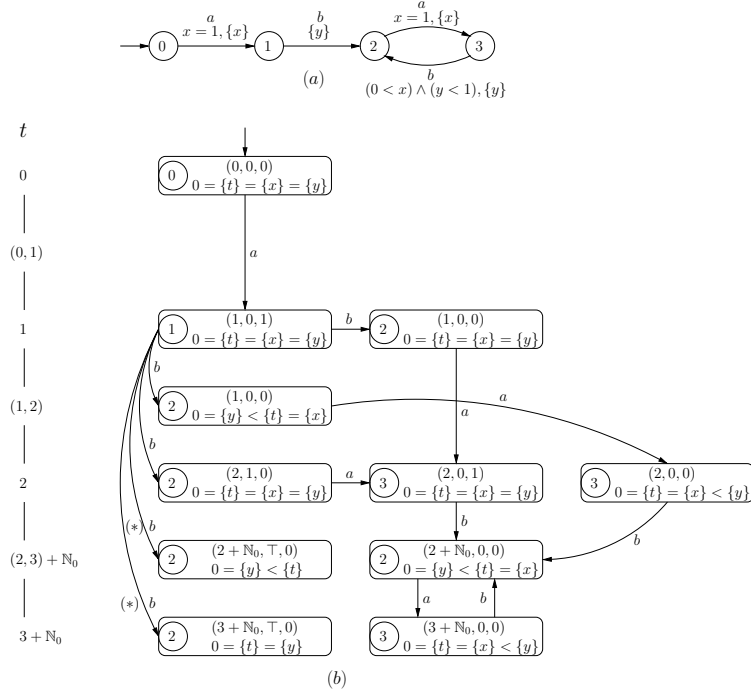


Figure 5: a)  $A \in \text{TA}$  ; b)  $\mathfrak{R}_{per}^t(A)$ , a periodic augmented region automaton of  $A$

## 6.1 Complexity

An upper bound on the number of nodes in the augmented region automaton  $\mathfrak{R}^t(A)$  is

$$N = O(|\mathcal{Q}|M^{|\mathcal{C}|}|\mathcal{C}|!), \quad (6)$$

where  $|\mathcal{Q}|$  is the number of locations,  $O(M^{|\mathcal{C}|})$  is the number of different integral values of the regular clocks, and  $O(|\mathcal{C}|!)$  is the number of different simplexes. The big-O notation refers here to a small factor. The actual number of nodes may be much smaller.

Let us look now at the number of nodes in  $\mathfrak{R}_{per}^t(A)$ . At each time level the number of nodes is bounded by  $N$ . Since  $t_0 \leq MN$  then until time  $t_0$  we have at most  $MN^2$  nodes. After passing  $t_0$  we have the subgraphs  $\bar{G}_k$  of time length  $L$ , where  $L$  is the period. Each such subgraph has at most  $NL$  nodes. Since the number of nodes in the subgraphs forms an almost increasing sequence (until an equality occurs two consecutive times), the number of nodes from time  $t_0$

to time  $t_{per}$  is bounded by  $(NL)^2$ . Thus, the number of nodes in  $\mathfrak{R}_{per}^t(A)$  is bounded by

$$O(L^2 + M)N^2). \quad (7)$$

The big factor in 7 may be the period  $L$ , so let us compute a bound on it.  $L$  is the LCM of the numbers  $d(\pi)$ , where  $d(\pi) \leq MN$  since the duration of each node is bounded by  $M$  and the length of a simple cycle is bounded by the number  $N$  of nodes in  $\mathfrak{R}^t(A)$ . Thus, a bound on  $L$  is given by the least common multiple of  $1, 2, \dots, MN$ , which is by the prime number theorem

$$L \leq \text{lcm}(1, 2, \dots, MN) = e^{MN(1+o(1))} \quad (8)$$

as  $MN \rightarrow \infty$ .

**Example 6.2.** When computing the period  $L$ , in the worst case the numbers  $d(\pi)$  are pairwise prime and the nodes of the cycles  $\pi$  form a disjoint union of sets which (almost) covers the set of nodes of  $\mathfrak{R}^t(A)$ . So, suppose that  $\mathfrak{R}^t(A)$  is in the form of  $n$  simple cycles, where each cycle is connected to the initial node by an additional edge. Suppose also that the length of cycle  $i$  is  $p_i$ , the  $i$ -th prime number,  $i = 1, \dots, n$ . Let us assume that  $M = 1$  and each edge is of weight 1. The number of nodes in  $\mathfrak{R}^t(A)$  is  $N = 1 + \sum_{i=1}^n p_i \sim (1/2)n^2 \log n$ . Since each cycle in  $\mathfrak{R}^t(A)$  is of type  $wZ1$ , we have  $L = \text{lcm}\{p_1, \dots, p_n\} = \prod_{i=1}^n p_i = e^{n \log n(1+o(1))}$ , the primorial  $p_n\#$ . This upper bound is closer to  $e^{M\sqrt{N}}$  than to  $e^{MN}$  of (8).

## 7 The Timestamp

The timestamp of the timed automaton  $A$ , denoted  $\mathfrak{TS}(A)$ , is the union of the timestamps of all observable transitions of  $A$ , that is, the set of all pairs  $(t, a)$ , such that an observable transition with action  $a$  occurs at time  $t$  in some run of  $A$ . By Theorem 5.3 and Proposition 3.2 we have the following.

**Theorem 7.1.** *The timestamp of a timed automaton  $A$  is a union of action-labeled integral points and open unit intervals with integral end-points. It forms an eventually periodic (with respect to time  $t$ ) subset of  $\mathbb{R}_{\geq 0} \times \Sigma$ , which can be effectively computed.*

**Example 7.1.** *The timestamp of the  $a$ -transitions of the automaton of Fig. 5 is  $\mathfrak{TS}^a(A) = \mathbb{N}$ , and that of the  $b$ -transitions is  $\mathfrak{TS}^b(A) = [1, \infty)$ .*

The timestamp of a timed automaton is an abstraction of its language. It does not preserve the timestamps of single timed traces. However, the timestamp is eventually periodic and computable, hence the timestamp inclusion problem is decidable. Thus, due to the general undecidability of the language inclusion problem in non-deterministic timed automata, one may use the timestamp for refutation purpose.

**Proposition 7.2.** *Given two eNTAs  $A$  and  $B$  then  $\mathfrak{TS}(A) \not\subseteq \mathfrak{TS}(B)$  implies  $\mathfrak{L}(A) \not\subseteq \mathfrak{L}(B)$ .*

Next we will show how to construct a *timestamp automaton*  $TSA(A)$  associated with the given timed automaton  $A$ . It is a simple deterministic TA with a single clock  $x$  and with timestamp identical to that of  $A$ . The graph of the timestamp automaton  $TSA(A)$  has the form of a ‘bouquet of flowers’. It is the disjoint union, except for the common initial location, of the subgraphs of the timestamps automata  $TSA(A_i)$ , where  $A_i$  refers to the automaton induced by  $A$  by transforming each transition in  $A$  into a silent transition, except for the transitions labeled by  $a_i$ . For each  $i$ ,  $TSA(A_i)$  is in the form of a simple prefix path (the ‘stalk’) which possibly ends with a simple loop (the ‘flower’), and we make sure that each stalk is of positive length.

Each guard in the stalk of  $TSA(A_i)$  matches the next time interval corresponding to the next  $a_i$ -interval in the timestamp  $\mathfrak{TS}^{a_i}(A)$ . We distinguish between two cases.

Case (i). If the periodic part of  $\mathfrak{TS}^{a_i}(A)$  contains an  $a_i$ -transition that occurs at an integral time then we prolong the stalk until performing this transition, while resetting  $x$ , and then we start the loop. Also,  $x$  is reset again when finishing the loop (see Fig. 6 (a)).

Case (ii). When the periodic part does not include a transition on an integral time then, if necessary, we prolong the stalk so that when we enter the loop it is on non-integral time, and we also reset  $x$  on this transition. Then, all transitions within the loop can occur at integral times (see Fig. 6 (b)). The idea is that if we enter the loop at a fractional time, say  $c = 0.3$ , then all the next transitions will take place at times  $n + 0.3$ , but since  $c$  can be chosen to be any real number in  $(0, 1)$  then the set of all runs will cover the entire timestamp.

**Example 7.2.** Let  $A$  be a timed automaton with timestamp

$$\begin{aligned}\mathfrak{TS}^a(A) &= (1, 3] \cup \{5\} \cup (6 + ([0, 2) \cup \{3\} \cup (8, 18)) + 21\mathbb{N}_0) \times \{a\}, \\ \mathfrak{TS}^b(A) &= [0, 1] \cup (2, 4) \cup \{5\} \cup (6 + ((0, 1) \cup (1, 2) \cup (5, 6) \cup (8, 9)) + 10\mathbb{N}_0) \\ &\quad \times \{b\}, \\ \mathfrak{TS}^c(A) &= [1, 4] \cup \{6\} \cup (10, \infty) \times \{c\}.\end{aligned}$$

Then a possible timestamp automaton of  $A$  is given in Fig. 6.

**Example 7.3.** The language of the timed automaton  $A \in \text{eNTA}$  of Fig. 7 (a) is  $\mathcal{L}(A) = \{(t_0, a), (t_1, a), \dots, (t_n, a) \mid i < t_i < i+1, i \in \mathbb{N}_0\}$  (supposing all locations are ‘accepting’). The timestamp of  $A$  is  $\mathfrak{TS}(A) = \{(n, n+1) \mid n \in \mathbb{N}_0\}$ .  $A$  is not determinizable. The reason is that each transition occurs between the next pair of successive natural numbers. The guard of each such transition must refer to a clock which was reset on some previous integral time. But since all transitions occur on non-integral time, the only clock the guards can use is the clock reset at time 0, hence the guards need to refer to all natural numbers, which is impossible by the finiteness of the automaton. Nevertheless, the timestamp automaton associated with  $A$ , seen in Fig. 7 (b), is deterministic.

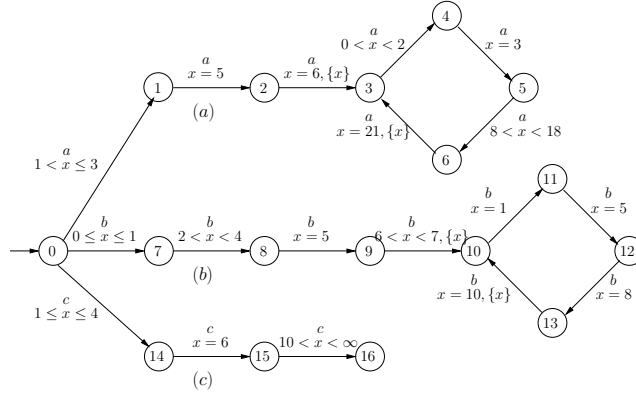


Figure 6: Timestamp automata of a)  $\mathfrak{TS}^a(A)$ ; b)  $\mathfrak{TS}^b(A)$ ; c)  $\mathfrak{TS}^c(A)$

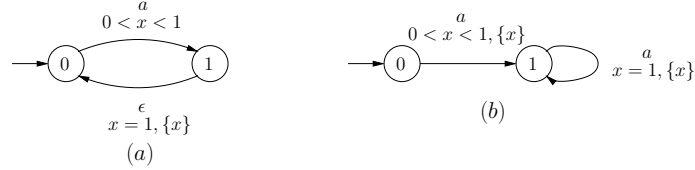


Figure 7: a) A non-determinizable  $A \in \text{eNTA}$  ; b) A timestamp automaton of  $A$

## 7.1 Timestamp of First Observable Event

We give here a simple application of the timestamp construction. The *universality* problem asks whether the language of a TA consists of all possible timed traces. It is known [AD94] to be undecidable in general. Here we address a simpler problem, which is decidable. Given a fixed positive integer  $k$ , we say that the language  $\mathfrak{L}(A)$  has the *k-bounded universality* property if  $\mathfrak{L}_k(A) := \{w \in \mathfrak{L}(A) \mid |w| \leq k\}$  contains all possible observable timed traces of length at most  $k$ .

**Proposition 7.3.** *The 1-bounded universality problem for eNTA is decidable.*

*Proof.* We add to  $A$  a ‘sink’ location  $q_{\text{sink}}$  and redirect each observable transition to it. Then we compute the timestamp of the connected component of the initial location and check whether it equals  $\mathbb{R}_{\geq 0}$ .  $\square$

In fact, the same construction for computing the first timestamp may be used for comparing  $A_1, A_2 \in \text{eNTA}$ . Let  $L = \text{lcm}(L_1, L_2)$ , where  $L_1, L_2$  are periods of  $A_1, A_2$  respectively, and let  $t_{\text{per}} = \max(t_{\text{per}_1}, t_{\text{per}_2})$ . Then we compute the first timestamp of  $A_1, A_2$  with period parameters  $t_{\text{per}}$  and  $L$ . Due to the eventual periodicity of the timestamps it suffices to conduct the comparison up to time

$t_{per} + L$ . We then have the following generalization of Proposition 7.3, which refers to the first observable transition.

**Theorem 7.4.** *The 1-bounded language inclusion problem for eNTA is decidable.*

Notice that the set of timed traces of observable length 1 may be unbounded by length or by time since each such timed trace may contain a prefix of silent transitions, and these prefixes may be unbounded in length or time due to cycles of silent transitions, also with clock reset. Hence, the bounded time [OW10] or bounded number of steps [LRNA15] techniques are not applicable here.

## 8 Conclusion and Future Research

The timestamp of a non-deterministic timed automaton with silent transitions (eNTA) consists of the set of all action-labeled times at which locations can be reached by observable transitions. The problem of computing the timestamp is a generalization of the basic reachability problem, a fundamental problem in model checking, thus being of interest from the theoretical as well as from the practical point of view. The periodic augmented region automaton contains more information than the timestamp, and it may also be equipped with guards on the edges so that the full information extracted from the original automaton is restored. As such, these constructions, that we have shown to be effectively computable, may assist us in comparing the languages of two timed automata, a task which is essential in testing and verification of abstract models of real-time systems, and which is, in general, an unsolvable problem.

We think that the importance of constructing the finite periodic augmented region automaton, besides being a tool in constructing the timestamp, lies in the explicit demonstration of the (global) time-periodicity of the timed automaton. For practical reasons, a future research may focus on constructing a similar time-periodic structure, but which is based on more compact representations such as zones or other symbolic representations [MPS11], instead of the more basic building blocks of regions.

When  $k > 1$ , the decidability of the  $k$ -bounded language inclusion problem for eNTA is, in general, still open, as far as we know. It cannot be handled by just considering the timestamp, and a more detailed exploration is needed, which takes into account the time durations between successive actions. This is another possible direction for future research.

## References

- [AD94] Rajeev Alur and David L. Dill. A theory of timed automata. *Theor. Comput. Sci.*, 126(2):183–235, 1994.

- [AFH99] Rajeev Alur, Limor Fix, and Thomas A. Henzinger. Event-clock automata: A determinizable class of timed automata. *Theor. Comput. Sci.*, 211(1-2):253–273, 1999.
- [AKV98] Rajeev Alur, Robert P. Kurshan, and Mahesh Viswanathan. Membership questions for timed and hybrid automata. In *Real-Time Systems Symposium*, pages 254–263, 1998.
- [AM99] Eugene Asarin and Oded Maler. As soon as possible: Time optimal control for timed automata. In *HSCC*, pages 19–30, 1999.
- [AM04] Rajeev Alur and P. Madhusudan. Decision problems for timed automata: A survey. In *Formal Methods for the Design of Real-Time Systems, International School on Formal Methods for the Design of Computer, Communication and Software Systems*, pages 1–24, 2004.
- [BBBB09] Christel Baier, Nathalie Bertrand, Patricia Bouyer, and Thomas Brihaye. When are timed automata determinizable? In *ICALP (2)*, pages 43–54, 2009.
- [BDFP04] Patricia Bouyer, Catherine Dufourd, Emmanuel Fleury, and Antoine Petit. Updatable timed automata. *Theor. Comput. Sci.*, 321(2-3):291–345, 2004.
- [BDM<sup>+</sup>98] Marius Bozga, Conrado Daws, Oded Maler, Alfredo Olivero, Stavros Tripakis, and Sergio Yovine. Kronos: A model-checking tool for real-time systems. In *CAV*, pages 546–550, 1998.
- [Bea98] Danièle Beauquier. Pumping lemmas for timed automata. In *FoS-SaCS*, pages 81–94, 1998.
- [BPDG98] Béatrice Bérard, Antoine Petit, Volker Diekert, and Paul Gastin. Characterization of the expressive power of silent transitions in timed automata. *Fundam. Inform.*, 36(2-3):145–182, 1998.
- [CG00] Christian Choffrut and Massimiliano Goldwurm. Timed automata with periodic clock constraints. *Journal of Automata, Languages and Combinatorics*, 5(4):371–404, 2000.
- [CHKM11] Taolue Chen, Tingting Han, Joost-Pieter Katoen, and Alexandru Mereacre. Reachability probabilities in markovian timed automata. In *CDC-ECC*, pages 7075–7080, 2011.
- [CJ99] Hubert Comon and Yan Jurski. Timed automata and the theory of real numbers. In *CONCUR*, pages 242–257, 1999.
- [CY92] Costas Courcoubetis and Mihalis Yannakakis. Minimum and maximum delay problems in real-time systems. *Formal Methods in System Design*, 1(4):385–415, 1992.

- [DT98] Conrado Daws and Stavros Tripakis. Model checking of real-time reachability properties using abstractions. In *TACAS*, pages 313–329, 1998.
- [Fin06] Olivier Finkel. Undecidable problems about timed automata. In *FORMATS*, pages 187–199, 2006.
- [HOW12] Christoph Haase, Joël Ouaknine, and James Worrell. On the relationship between reachability problems in timed and counter automata. In *Reachability Problems*, pages 54–65, 2012.
- [HP06] Thomas A. Henzinger and Vinayak S. Prabhu. Timed alternating-time temporal logic. In *FORMATS*, pages 1–17, 2006.
- [LPY97] Kim G. Larsen, Paul Pettersson, and Wang Yi. Uppaal in a nutshell. *STTT*, 1(1-2):134–152, 1997.
- [LRNA15] Florian Lorber, Amnon Rosenmann, Dejan Nickovic, and Bernhard K. Aichernig. Bounded determinization of timed automata with silent transitions. In *FORMATS*, pages 1–17, 2015.
- [MPS11] Georges Morb , Florian Pigorsch, and Christoph Scholl. Fully symbolic model checking for timed automata. In *CAV*, pages 616–632, 2011.
- [ORW09] Jo l Ouaknine, Alexander Rabinovich, and James Worrell. Time-bounded verification. In *CONCUR*, pages 496–510, 2009.
- [OW04] Jo l Ouaknine and James Worrell. On the language inclusion problem for timed automata: Closing a decidability gap. In *LICS*, pages 54–63, 2004.
- [OW10] Jo l Ouaknine and James Worrell. Towards a theory of time-bounded verification. In *ICALP (2)*, pages 22–37, 2010.
- [TY01] Stavros Tripakis and Sergio Yovine. Analysis of timed systems using time-abstracting bisimulations. *Formal Methods in System Design*, 18(1):25–68, 2001.
- [Wan04] Farn Wang. Efficient verification of timed automata with bdd-like data structures. *STTT*, 6(1):77–97, 2004.
- [WZP03] Bozena Wozna, Andrzej Zbrzezny, and Wojciech Penczek. Checking reachability properties for timed automata via SAT. *Fundam. Inform.*, 55(2):223–241, 2003.